The system **100** can do so by determining (**503**) whether context parameter **2** has a value of "Y," which can include a parameter such as the "market segment" associated with the user. If the value of context parameter **2** is not "Y," then policy instance **2** can be associated (**504**) with the particular user. If the system **100** determines that context parameter **2** does have a value of "Y," then the system **100** can determine (**505**) another potential policy instance override point. The system **100** can determine (**505**) whether context parameter **3** has a value of "Z," which can include a parameter such as "client." If the system **100** determines that the value of context parameter **3** is not equal to "Z," then policy instance **3** is associated (**506**) with the particular role. If the system **100** determines that context parameter **3** is equal to "Z," then policy instance **4** is associated (**507**) with the particular user. Hierarchical context parameters, for example, context parameters **1**, **2**, and **3** of FIG. **5**, can play a part in the customization of policy instances associated with the role of a particular user. In this example, the customizable policy instances can facilitate reduction in the number of roles that have various authorizations to resources. Such facilitation can be made possible when the system **100** takes into account the one or more unique context parameters for a given user in a particular role rather than requiring that a systems administrator define numerous roles to accommodate each of the possible contextual scenarios within which the user may operate.

[0045] If a policy type, for example, "compensation," is expanded to accommodate additional functions, features, or capabilities, such as updating a salary amount for an employee, the expansion is applied to all policy instances based upon the policy type. A systems administrator may define, for example, such new policy element values at the default policy instance level. This expanded definition can automatically apply to all subordinate policy instances.

[0046] FIG. **6** is an exemplary illustration for role-based access **600** in a multi-customer computing environment. A user **110** (one type of actor) logs on to a computer system by providing, for example, a userID and password, and attempts interaction with one or more product modules **602**. A product module **602** can represent one or more applications with which a user **110** attempts interaction. The identity of the user **110** is established through an authentication module **604** using the authentication credentials that the user **110** inputs into the system. If the user **110** is authenticated by the authentication module **604**, then one or more predefined context parameters are obtained, e.g., "market segment A." Context parameters can be, for example, cached or obtained from existing databases within, for example, one or more recordkeeping systems. The illustrated communications can be performed using one or more standards-based languages and protocols, such as SOAP, HTTP, IP, SQL, and the like.

[0047] A user **110** is assigned to a role based on, for example, the identity of the user and the one or more pre-defined context parameters. Role assignment can be accomplished, for example, by a role resolution application module **608***a* based on values obtained from, for example, an actorID-role database **606***a*. In this example, the role of "manager" is assigned based on the userID and the "market segment A" context parameter.

[0048] A role scope can be associated with a role. Such associations can be stored, for example, in a role scope-role database **606***b*. The role scope includes a general specification of the scope associated with each role. In this example, the role of "manager" has a role scope that includes "orgID," "department name," and "direct reports." The actor-role scope key for this example includes "org22" as the orgID, and "R&D" as the department name. These actor-role scope key values can represent one or more pointers that enable the resolution of scope, which includes the populations and/or entities that a user **110** may act upon. In this example, the resolved scope represents the direct reports "E. Smith," "F. Jones," and "T. Roberts."

[0049] A policy type is associated with the role of a user **110**, which can be, for example, stored in a policy type-role database **606***c*. In this example, the policy type "time keeping" is associated with the role of "manager." The "time keeping" policy type includes, for example, the "enter time keeping data," and the "approve time keeping data" policy elements.

[0050] Existing context parameters are read and/or additional context parameters are obtained in order to map one or more policy instances to a role. Based on the role of "manager," and the context parameter "market segment A," a policy instance for the policy type "time keeping" is mapped **608***b* to the "manager" role. In this example, the policy element values for "enter time keeping data" and "approve time keeping data" are both set to "yes." Based on the policy instance that was mapped **608***b* to the "manager" role, the manager in this example is authorized to enter and/or approve time keeping data for E. Smith, F. Jones, and T. Roberts.

[0051] The role and a policy instance of an access control policy element can determine what the system makes available to a given user **110**. The policy instance for a data view/presentation policy element can define which aspects of a system component or entity are to be presented to a user **110**. For example, a "manager" role and a "human resource specialist" role may have access to the same system components or entities, e.g., personnel information, based on the policy instance for the access control policy element (named "access personnel data") of the policy type (entitled "performance management") associated with each of their roles. The "manager," however, may have rights to view employee salaries based on the policy instance for the data view policy element (named, for example, "view personnel data") associated with the policy type (entitled, for example, "performance management") for his or her role, whereas the "human resources specialist" may not possess such rights. The instance associated with a function performance/update operation policy element can control what functions the user **110** is allowed to perform and those he or she may not perform when the user **110** interacts with the system **100**. The "manager" role may, for example, be able to effect a promotion by, among other tasks, modifying the personnel data associated with an employee's title. Whereas, the "human resources specialist," having no such promotion authority, will not have modification rights to employees' titles. This distinction in update authority is based upon the different policy instances for the function performance/update operation policy elements associated with the respective policy types and roles for the "manager" and the "human resources specialist."

[0052] FIG. **7** illustrates another example in which context parameter-based roles and policy instances can determine the one or more resources that a user may be authorized to access. FIG. **7** illustrates users **110** in the role of "manager"